REMARKS

Reconsideration of this application, as amended, is respectfully requested. New claims 35-37 are supported by the specification as originally filed. For example new claim 35 finds support in paragraph 89 of the specification. New claim 36 recites subject matter similar to that of original claim 4. New claim 37 finds support in paragraph 86 of the specification. Hence, no new subject matter is introduced by the addition of these claims.

Claim 1 has been amended to clarify that the request originates at a client and the address of the selected information object repository is returned to the client. No new matter is introduced by these clarifying amendments, which were made solely to better illustrate the points discussed below.

1. The Present Claims are Patentable over the Combination of Jordan and Pistriotto

In rejecting claims 1-28 and 30-34, the Final Office Action relies on the combination of Jordan (US 6438652) and Pistriotto (US 6138162).[1] For the reasons set forth below, these rejections are respectfully traversed. The combination of these references: (1) fails to teach or suggest features of the present claims, and (2) when construed properly teaches a redirection scheme different from that presently claimed. Accordingly, the present claims should be deemed patentable over the combination of these references.

Jordan describes a redirection scheme in which a central load balancer (120) is used to orchestrate redirection decisions made by a number of cooperating cache servers (150). The redirection scheme operates as follows:

> First, a client browser (160) makes a request to one of the cooperating cache servers for an object. If the object is stored locally, the cache that received the client's request returns the object to the client. If, however, the cache does not

[1] Various dependent claims are rejected based on the teachings of additional references considered in combination with Jordan and Pistriotto, however, it is the Jordan and Pistriotto references that are relied upon as the basis for the rejection of all parent claims of these dependent claims.

have a copy of the object, it sends a request for that object to the load balancer. See Jordan at col. 7, ll. 35-42.

Second, and depending on load conditions amongst the cooperating caches, the load balancer determines which cache should respond to the original cache server's request. Specifically, the load balancer forwards the original cache's request to the new cache server. Jordan at col. 7, ll. 51.

Third, the new cache server that receives the forwarded request from the load balancer returns the requested object to the cache that made the request. In some cases, this may require the new cache to actually obtain a copy of the requested object. Jordan at col. 7, ll. 5-7. Thereafter, the cache that received the original client request can now return the requested object to the client.

Jordan discusses various load balancing algorithms that can be used by the load balancer in deciding where to redirect requests after cache misses, but importantly, the load balancer always forwards requests to the cache server that is to respond rather than instructing the original cache to make such contact on its own. Stated differently, Jordan does not return to the requesting cache or any other entity an address of a selected cache that will fulfill the request.

Turning now to Pistriotto, a very different sort of redirection scheme from that found in Jordan is discussed. In the Pistriotto system, before any redirection occurs, the client must contact the origin server to obtain a list of cache sites from which the client may request various forms of objects or services. Pistriotto at col. 7, l. 49 – col. 8, l. 2. After such a list is obtained (and which list must be periodically validated) the client sends direct requests to a designated proxy cache for content.

It is questionable whether a mechanism such as that described by Pistriotto would ever have been considered for use in combination with the system taught by Jordan. For example, the Jordan load balancing system is designed to relieve burdens on origin

servers (and other cache servers) by redirecting requests dynamically according to various load balancing criteria. The Pistriotto scheme, however, uses a fixed table to instruct a client to always contact a specific cache for designated types of content. Moreover, in the Pistriotto system the origin server must continually update the tables and provide them to the clients and so it is questionable how much burden of responding to client requests is actually relieved.

Because there appears to be little or no motivation for making the combination relied upon by the examiner then, the rejection of the claims in light of this combination is flawed.

Even if the combination of the references is proper, the result does not obviate the presently claimed method. Notice that the request made by a client in the Pistriotto system is NOT a request for an information object (as recited in claim 1). Instead, it is a request for a redirection table. Hence, if one were to adopt the techniques taught by Pistriotto in the system described by Jordan, requests for information objects would not be treated any differently than Jordan already describes. Stated differently, Pistriotto does not suggest modifying the manner in which Jordan treats requests for information objects, only (perhaps) modifying the way Jordan teaches requests for redirection instructions.

Consequently, in a system based on the combination of the Jordan and Pistriotto teachings, requests for information objects would go first to a cache, and, upon a cache miss, would be referred to a load balancer and forwarded to a second cache according to the load balancing algorithms taught by Jordan. Importantly, neither the requesting client nor even the cache receiving that request would be provided with the address of the second cache to which the request was referred. Stated differently, the load balancer would not return to the client (or the original cache for that matter) an address of a selected information object repository (i.e., the second cache).

For at least these reasons, claim 1 is patentable over the combination of Jordan and Pistriotto. Moreover, because all of the remaining claims in this application depend from claim 1 and therefore include the features thereof, those claims too are patentable over tis combination of references.

2. Claims 8-10, 21-28, 31 and 34 are Patentable over Jordan and Pistriotto even when Considered in Combination with Rune.

From the above it should be apparent that claims 8-10, 21-28, 31 and 34 are patentable over the combination of Jordan and Pistriotto. Rune (US 6304913) fails to cure this deficiency.

Rune describes a method for selecting a closest server based on hop counts, which may be computed in different ways. Rune at col. 4, ll. 29-47. Rune does not, however, teach or suggest a redirection scheme in which the address of a selected information object repository is returned in response to a request for an information object from a client. Accordingly, even if Rune does discuss methods for determining a topologically close cache, such teachings are irrelevant because the claims are patentable for other reasons.

3. Claims 16-20 are Patentable over Jordan and Pistriotto even when Considered in Combination with Chauhan.

From the above it should be apparent that claims 16-20 are patentable over the combination of Jordan and Pistriotto. Chauhan (EP 0959601) fails to cure this deficiency. Chauhan describes a method for selecting a server from among a number of mirrors. Such teachings are irrelevant, however, because the claims are patentable for other reasons as discussed above with respect to claim 1.

4. Claims 11-15, 30, 32 and 33 are Patentable over Jordan and Pistriotto even when Considered in Combination with Johnson.

From the above it should be apparent that claims 11-15, 30, 32 and 33 are patentable over the combination of Jordan and Pistriotto. Johnson (US 6205477) fails to cure this deficiency. Johnson describes a method for distributing service requests among a number

of servers according to position metrics. Such teachings are irrelevant, however, because the claims are patentable for other reasons as discussed above with respect to claim 1.

For at least the foregoing reasons, the claims are patentable over the cited references. If there are any additional charges, please charge Deposit Account No. 02-2666.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Dated: 12/17, 2004

Tarek N. Fahmi
Registration no. 41,402

12400 Wilshire Blvd.
Seventh Floor
Los Angeles, CA 90025-1026
(408) 947-8200